Barea, R., Bergasa, L. M., Romera, E., López Guillén, E., Pérez, O., Tradacete, M. & López, J. 2019, "Integrating state-of-the-art CNNs for multi-sensor 3D vehicle detection in real autonomous driving environments", en 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, pp. 1425-1431

Available at http://dx.doi.org/10.1109/ITSC.2019.8916973

Integrating State-of-the-Art CNNs for Multi-Sensor 3D Vehicle Detection in Real Autonomous Driving Environments

Rafael Barea[1], Luis M. Bergasa[1], Eduardo Romera[1], Elena López-Guillén[1], Oscar Perez[1], Miguel Tradacete[1], Joaquín López[2]

***Abstract*— This paper presents two new approaches to detect surrounding vehicles in 3D urban driving scenes and their corresponding Bird's Eye View (BEV). The proposals integrate two state-of-the-art Convolutional Neural Networks (CNNs), such as YOLOv3 and Mask-RCNN, in a framework presented by the authors in [1] for 3D vehicles detection fusing semantic image segmentation and LIDAR point cloud. Our proposals take advantage of multimodal fusion, geometrical constrains, and pre-trained modules inside our framework. The methods have been tested using the KITTI object detection benchmark and comparison is presented. Experiments show new approaches improve results with respect to the baseline and are on par with other competitive state-of-the-art proposals, being the only ones that do not apply an end-to-end learning process. In this way, they remove the need to train on a specific dataset and show a good capability of generalization to any domain, a key point for self-driving systems. Finally, we have tested our best proposal in KITTI in our driving environment, without any adaptation, obtaining results suitable for our autonomous driving application.**

I. INTRODUCTION

One of the main problems when designing perception systems for autonomous driving is object detection in 3D space. An autonomous vehicle needs to localize and track their surrounding obstacles from its own sensors in order to plan its path in a safe way. Nowadays, most self-driving vehicles are geared up with multiple high-precision sensors such as LIDAR and cameras.

LIDAR-based detection methods provide accurate depth information and obtain robust results in location, independently of the environment lighting conditions. However, these approaches struggle at long range and when dealing with occluded objects due to the sparsity of the LIDAR point samples over these regions [2]. On the other hand, camera-based methods provide much more detailed semantic information. However, their performance degrades in situations with challenging lighting conditions (e.g. sun glares, dark scenes) and with distance. Besides, precise 3D localization using only monocular cameras is hard to achieve due to the loss of depth information, which must be recovered by projection models that inevitably increase uncertainty in the distance estimation. Recently, several works have proposed methods that exploit LIDAR and cameras complementarily to alleviate drawbacks present in the respective individual modalities, thus achieving higher performance [3]. However, the challenge lies in the applied fusion technique, especially considering that LIDAR points are sparse and continuous, while cameras capture dense features at a discrete state [2].

In recent years, Convolutional Neural Networks (CNNs) have reached great success in object detection achieving the top ranked results on public benchmarks such as KITTI [4]. 2D detection from images has seen significant progress [5], [6], [7]. However, there is still large room for improvement in the 3D object detection case. Some proposals give 3D pose estimation from solely monocular RGB images, as in [8]. Others estimate 3D object detection directly on LIDAR point clouds [9], [10], or convert point cloud data into a 2D Bird's Eye View (BEV) [11], [3]. Recent approaches exploit both cameras and LIDAR jointly [10], [11], [12], [2].

All the above proposals use end-to-end supervised learning trained on the KITTI dataset, and their main research efforts are being invested on designing and enlarging deep architectures to achieve marginal accuracy boosts in specific datasets, neglecting that these algorithms must be run in a real vehicle with constrained computational devices and must work robustly in diverse image domains that weren't seen in the training process (which can be very different in perspective and appearance). After all, CNNs are trained on a limited dataset, and there is no guarantee that the latent representation learned from it is transferred properly to any domain [13].

In this paper, we aim to leverage two state-of-the-art pre-trained CNNs thought to produce accurate 2D object detection in images, such as YOLO [5] and Mask-RCNN [7], to detect vehicles in a 3D scene with the aim of generating a safety path in a real autonomous driving application for urban environments. We take as a starting point our previous work [1], where a framework for 3D vehicle detection fusing semantic image segmentation, through our ERFNet (Efficient Residual Factorized ConvNet), and LIDAR point cloud [14] was proposed. In this work we analyze two improvements with respect to our baseline: 1) Use a YOLOv3 to get the 2D box proposals on the image instead of using the vehicle class as priors, 2) Substitute our ERFNet for a Mask-RCNN in order to get a semantic segmentation of the 3D box proposals

*This work has been funded in part from the Spanish MINECO/FEDER through the SmartElderlyCar project (TRA2015-70501-C2-1-R, TRA2015-70501-C2-2-R), and from the RoboCity2030-DIH-CM project (P2018/NMT-4331), funded by Programas de actividades I+D (CAM) and cofunded by EU Structural Funds.

[1] Rafael Barea, Luis M. Bergasa, Eduardo Romera, Elena López-Guillén, Oscar Perez and Miguel Tradacete are with the Electronics Department, University of Alcalá (UAH), Spain {rafael.barea, luism.bergasa, elena.lopezg}@uah.es,{eduardo.romera, oscar.perez, miguel.tradacete}@edu.uah.es

[2] Joaqín López is with the Department of Systems Engineering and Automation, University of Vigo, Pontevedra, Spain joaquin@uvigo.es

(mask) and the 2D box proposals (R-CNN) with the same network.

In our experiments, the performances of the two newly proposed frameworks are extensively tested on the KITTI object detection benchmark [4] as well as on sensor data fully captured by our autonomous electric vehicle, which has been manually annotated by us to enhance our experimental test set and to provide quantitative results in a real environment. Results show that our proposals are on par with other references of the state of the art and present a good generalization capability, a key point for real autonomous navigation applications. Furthermore, our approach successfully integrates robust and well-known architectures that were pretrained on generalistic datasets, removing the need to train our framework in specific and limited datasets like KITTI (which are easy to overfit) and avoiding the costly efforts of annotating new training data.

II. Related Works

In this section we briefly review the most relevant works of the state of the art about object detection based on LIDAR point clouds, camera images, and their fusion.

A. LIDAR-based Detection

The majority of existing methods encode 3D point clouds in voxel grid representations and rely on basic features for classification. Some works, such as Vote3D [15], use SVM classifiers on 3D clusters encoded with geometry features. Other works, such as [9], [10], propose to improve feature representation by using 3D convolutional networks directly on the LIDAR point clouds. VeloFCN [16] projects the LIDAR points to front view and applies a 2D fully convolutional network to generate 3D detections. In [11], [3] the same strategy is used, but projection is carried out on the ground plane, generating a 2D BEV. PIXOR [17] exploits a height-encoded bird's-eye view representation of the LIDAR and applies a 3D fully convolutional network.

Most algorithms discretize point clouds into a 3D grid. Point density decreases with distance and classifiers must deal with both dense and sparse points in the same scene. In practice, classifiers usually work well in short range, where dense points are available, and hardly work properly in long-distance, where only sparse points are available. Our framework provides semantic information to the point cloud to improve 3D classification, specially at long-distance [1]

B. Image-based Detection

In the last years, many methods that exploit convolutional neural networks have played an important role in producing accurate 2D object detection, typically from a single image. There are two different approaches depending on their stage detection framework.

1) One-stage detectors: learn a network that directly produces object bounding boxes. Notable examples are YOLO [5] and SSD [6]. They are computationally attractive and run in real time on most hardware.

2) Two-stage detectors: utilize region proposal networks in a first stage to learn the region of interest (RoI) where potential objects are located. In a second stage, the CNN is applied on the detected RoIs to classify each object and refine its location.

Faster R-CNN [18] is one of the most popular methods for vehicle detection. In the 3DOP method [19] the 3D box proposals are fed to an R-CNN pipeline to detect vehicles from stereo images. Mono3D method is presented in [8]. It uses the same pipeline that 3DOP but in this case, it generates 3D proposals from monocular images. Mask-RCNN [7] also takes this approach, but it addresses the boundary and quantization effect of RoI pooling in the 2D image detection stage. Besides, it adds an additional segmentation branch to take advantage of dense pixel-wise supervision, providing instance segmentation through a mask.

The main drawback of both of these image-based methods becomes apparent in the fusion with 3D information. Obtaining a 3D depth estimation in images usually relies on a well-calibrated camera model, which in practice is never as accurate as LIDAR-based estimations. Therefore, our framework incorporates LIDAR point clouds to improve 3D localization.

C. Multimodal Fusion

Multiple data fusion provides complementary information, therefore increasing the decision-making accuracy in self-driving systems [11]. Over the past few years, many proposals have explored merging both cameras and LIDAR to perform 3D reasoning. In [10], proposals are generated from images and LIDAR is used to conduct the final 3D localization. This approach does not exploit the capability to perform joint reasoning over the two inputs. Another approach [12] applies 2D convolutional networks on both camera images and a LIDAR BEV representation and fuses them at a course level of the features map with significant resolution loss. [11] introduces a Multi-View object detection network (MV3D) to fuse features from multiple sensors in multiple views through RoI-pooling. Accurate geometric information is lost in this coarse pooling scheme. UberATG [2] proposes a 3D object detector that reasons in BEV and fuses image features by learning to project them into BEV space.

Our approach uses multimodal fusion but differs from previous approaches in that our fusion method does not require learning an end-to-end strategy but a high level geometric one based on the proposals obtained from the two sensors.

III. FRAMEWORK IMPROVEMENTS

In this section, we explore and describe two framework options to improve the performance in the task of Multi-Sensor 3D Vehicle Detection, carried out by taking our previous framework [1] as a baseline.

In the former framework, 2D proposals on the image were calculated using the blobs belonging to a certain class obtained from the semantic segmentation through our ERFNet.

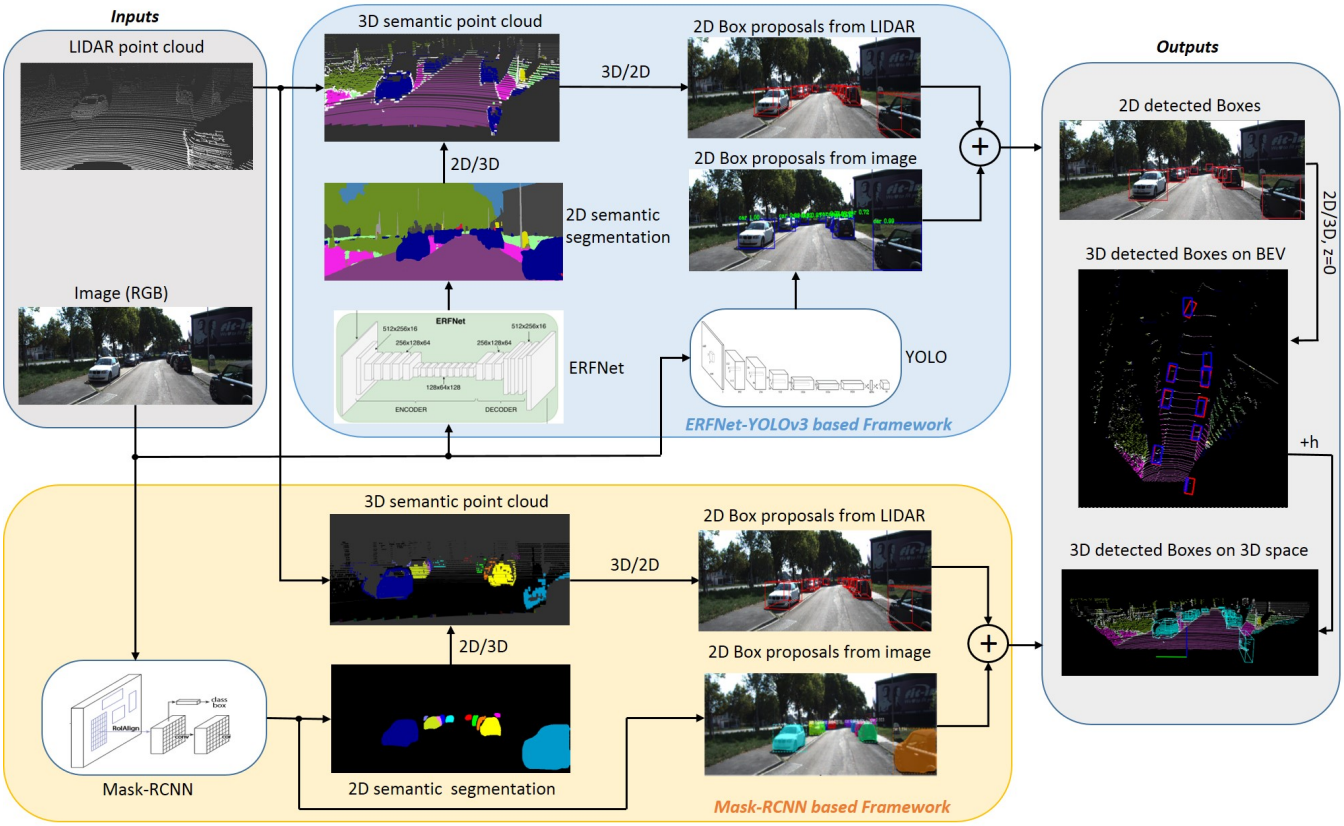


Fig. 1: Framework based architectures.

The problem is that contiguous instances of a class were detected as only an object in the image. To solve this problem we propose using a YOLO object detector as a 2D vehicle proposal generator. On the other hand, using different CNNs (with different feature extractors) for semantic segmentation and for object detection can be considered suboptimal in terms of framework design. This is why we propose the second framework option (as an alternative), by replacing our ERFNet for a Mask-RCNN in order to get the semantic segmentation for the 3D box proposals and the 2D box proposals using only one network.

A. ERFNet-YOLOv3-based Framework

This approach takes data from LIDAR and RGB images. Fig.1 shows an overview diagram for this architecture.

1) ***2D object detection from RGB image****:* Object detection is carried out by processing the RGB image with YOLOv3, which was pre-trained in the COCO dataset and did not require any additional training or adaptation to this new framework. A comparison with other detectors of the literature shows that YOLOv3 is extremely fast and accurate [5], requirements for a vehicle detector system suitable to be used in autonomous driving. YOLO provides a set of 2D proposals which encode probabilities and bounding box information for detected objects in the image.

2) ***Semantic segmentation****:* In parallel to our object detection using YOLOv3, our ERFNet obtains the semantic segmentation of the RGB image. This network is a deep architecture able to provide accurate semantic segmentation running in real-time. The core of our Net is a novel layer based on residual connections and factorized convolutions to retain remarkable accuracy while remaining efficient [14]. The ERFNet was trained on Cityscapes [20] with 19 classes because our goal was to achieve robustness in any domain.

3) ***3D object detection from LIDAR colored point cloud****:* Given the LIDAR point cloud and the semantic information a 3D colored point cloud is obtained, where different objects in the 3D scene are classified by color (see Fig.1). To do that, each point of the cloud is projected to the semantic image using an algorithm based on [21], and colored according to the color of the object class on which it is projected. Due to points with the same color belonging to the same class, classification is carried out by color filtering. However, different objects belonging to the class are connected in some cases (see cars in Fig.1) and additional processing is required to separate them. A clustering based on Euclidean distance, as proposed by [22], is carried out over the point cloud with the same color to detect the different objects in the scene for each class.

After that, a 3D bounding box that best suits the shape of each cluster is assigned to each object. Length of the boxes are discretized to 5 different values and height is fixed to 1.6m, according to the mean clusters for length/height obtained for the car class on KITTI. 3D bounding box fitting is very sensitive to orientation, which is difficult to estimate due to occlusions and sparsity of data. To improve bounding box pose estimation (position and orientation), we project orthogonally the 3D point cloud to the 2D ground plane (z=0) and fit a 2D box to each object using the Hough transform to

get the main box directions, as we explained in our previous publication [1].

*4) **Fusion from LIDAR and image proposals**:* 3D LIDAR box proposals and 2D image box proposals are merged to validate common detection and to complement detection carried out for one of the sensors only. To do that, 3D box proposals are projected to the image plane. In this way, 2D box proposals coming from the LIDAR point cloud are easily matched with the 2D box proposals coming directly from the semantic image. If a proposal overlaps (IoU) in the two domains and deals some geometric constraints, it is validated and it is the LIDAR proposal who goes to the output detected vehicles image. On the other hand, if a proposal appears only in one domain it is validated and goes to the output image depending on the sensor (LIDAR or image) and the distance where it was found. More information can be found in our previous work [1].

Finally, validated 2D boxes are projected back to the ground plane (BEV detection), and full 3D detection is achieved by introducing height templates to the different detected objects (see Fig.1).

B. Mask-RCNN-based Framework

This approach is similar to the previous one, but in this case we substitute the YOLOv3 and ERFNet by a Mask-RCNN (Region-based Convolutional Neural Network) [23] in order to simplify the architecture. Fig.1 shows an overview diagram of this architecture.

*1) **2D object detection from RGB image**:* The object detection task is implemented using the object detection branch of a Mask-RCNN. In this way, a set of 2D proposals which encode probabilities and bounding box information for objects detected in RGB image is provided.

*2) **Semantic segmentation**:* For each image, the Mask-RCNN implements a region proposal network which extracts a RoI, predicting a segmentation mask in a pixel-to-pixel manner. Results obtained in this segmentation process do not match exactly those obtained by the previous ERFNet; the performance and detected classes differ significantly between both cases since they are different architectures and they were trained in different datasets. In this case, only vehicles, bicycles and pedestrian classes are detected, but the network is able to distinguish the different instances for each class, given a different color for each of them. Fig. 1 shows the different vehicle instances, but unlike the previous approach, there is no information about the road and other elements of the environment.

*3) **Object detection from LIDAR colored point cloud**:* As in the previous case, the 3D point cloud is projected to the 2D semantic image obtained from Mask-RCNN instances. This way, a point of the cloud projected into an instance is colored according to its color. After that, classification is carried out by color filtering taking into account that in this case each color corresponds to a different object, even belonging to the same class (instance). This way a clustering is not necessary to get connected objects in the scene of the same class. Later, a 3D bounding box is fitted according to size and orientation of each object, in the same way as explained above. Finally, a 3D box proposal is obtained.

*4) **Fusion from LIDAR and Image proposals**:* The same method is applied as in the previous approach.

IV. EXPERIMENTAL RESULTS

A. KITTI Benchmark

We evaluate our 3D vehicle detection proposals on the challenging KITTI object detection benchmark [4]. The dataset provides 7,481 images for training with ground truth annotations and 7,518 images for online testing without ground truth. As the online testing only evaluates 2D detection, we conduct our evaluation on the training set. To evaluate localization, we use point cloud in the range of [0,70]x[-40,40] meters.

Evaluation is carried out on the whole KITTI training set for the ”car” class, taken into account that these images have not been seen before by CNNs implemented in our approaches. In other approaches in the literature, research efforts are often invested in designing large architectures to achieve accuracy boosts in KITTI by training and sometimes overfitting on its train set, which is very similar to the test set. This is problematic for a system that aims to be deployed in the real world, where the test domain will be completely different to the one that was planned in training. For these reasons, and since our goal is to achieve robustness in any domain, we restricted our framework design to avoid using any KITTI data for training.

We validate our proposal in both 2D/3D space using Average Precision (AP) with the following metrics. For 2D detection on the images, Intersection over Union (IoU) is used to distinguish between true positive and false positive with a threshold of 0.7. For 3D object detection, 3D IoU is applied with a threshold of 0.5. This metric shows the highest demand because 3D overlapping is evaluated. For BEV object detection, 2D IoU on BEV is used with the same threshold. In this case the metric shows autonomous driving demand, in which vertical localization is less important than the horizontal. KITTI divides the labels into three difficulty modes: easy, moderate and hard, according to the heights of their bounding boxes, truncation levels and occlusion levels. In the detection results tables, a RGB color code will be used to highlight the three highest values.

B. Baseline for Comparison

As this work aims at 2D/3D vehicle detection, for the 3D and BEV evaluation we compare our approach to a representative LIDAR-based method such as VeloFCN [16], representative image-based methods such as 3DOP [19] and Mono3D [8], as well as a reference of the multimodal methods (LIDAR + image) such as is the MV3D [11]. For 2D detection evaluation, we add Vote3D [15], YOLO [5] and Mask-RCNN [7].

C. Performance of 2D Vehicle Detection

2D detection performance for the car class and for an IoU=0.7 on the KITTI test set, except for our framework

approaches, can be found in Table I. In our cases, the whole training set is used due to it has not been used for training our CNNs. As can be seen, image-based methods perform better than LIDAR-based ones. The reason can be found in that image-based methods directly optimize 2D boxes while LIDAR-based ones optimize 3D boxes. Fusion proposals (MV3D) optimize both 2D/3D boxes and in consequence get intermediate results.

TABLE I: 2D Detection performance: Average Precision (AP) in % for car class on KITTI test set, excepts for our proposal where the whole training set was used. IoU=0.7

Method	Data	Easy	Mod.	Hard
Mono3D [8]	Mono	92.33	**88.66**	**78.96**
3DOP [19]	Stereo	**93.04**	**88.64**	**79.10**
YOLOv3 [5]	Mono	84.30	84.13	76.34
Mask-RCNN [7]	Mono	87.90	79.11	70.19
VeloFCN [16]	LIDAR	71.06	53.59	46.92
Vote3D [15]	LIDAR	56.80	47.99	42.57
MV3D [11]	LIDAR+Mono	89.11	**87.67**	**79.54**
Our ERFNet [14]	LIDAR+Mono	90.45	78.28	73.20
Our ERF+YOLOv3	LIDAR+Mono	**93.75**	83.79	76.32
Our Mask-RCNN	LIDAR+Mono	**94.07**	86.58	77.70

Our approaches outperform LIDAR-based for all test difficulty modes and are on par with the obtained by image-based method and MV3D fusion method, being a little better for the easy mode and a little worse for moderate mode and hard mode. New proposals improve results regarding the baseline in more than 3% for the easy mode, more than 5% for the moderate one and more than 3% for the hard one. Results using Mask-RCNN in our framework are a little better than those obtained by using ERF+YOLOv3. It is then remarkable that results obtained using only YOLOv3 and Mask-RCNN are worse than using these CNNs inside our fusion framework.

Our proposals show comparable results with other image-based and fusion-based methods of the state of the art that use end-to-end learning in KITTI, but in our case, we use CNNs as modules trained on the COCO dataset and the Cityscapes dataset, quite different to KITTI, showing a high capability to generalize domains. However, these results are not enough for autonomous driving applications where 3D vehicle detection is the key parameter.

D. Performance of 3D Vehicle Detection

3D IoU is the most precise metric to evaluate vehicle detection in a 3D autonomous driving scenario. However, in this context, vertical localization is less important than the horizontal one. This is the reason why most of the state-of-the-art methods predict the 2D height in a decoupled and coarse way (sometimes a fixed value is assigned for all detected objects of a class), provoking a negative impact in the performance numbers.

Table II shows AP on the KITTI validation set using a 3D IoU threshold of 0.5. The LIDAR-based method (VeloFCN) performs better than image-based methods (Mono3D, 3DOP) due to LIDAR sensors obtaining distance measurements directly. For the fusion proposals (MV3D and ours), best results for all test difficulty modes are obtained. MV3D performs much better than our proposals, but our methods are the only ones that have not been trained with KITTI images. Besides, we use a fix height of 1.6 m for all cars, because our real goal is to build a perception system for a real vehicle and not get top rank in KITTI benchmark.

TABLE II: 3D Detection performance: Average Precision (AP) in % of 3D boxes on KITTI validation set for IoU=0.5

Method	Data	Easy	Mod.	Hard
Mono3D [8]	Mono	25.19	18.2	15.52
3DOP [19]	Stereo	46.04	34.63	30.09
VeloFCN [16]	LIDAR	67.92	57.57	52.56
MV3D [11]	LIDAR+mono	**96.02**	**89.05**	**88.38**
Our ERFNet [14]	LIDAR+mono	78.09	60.23	55.48
Our ERF+YOLOv3	LIDAR+mono	**85.04**	**61.91**	**56.76**
Our Mask-RCNN	LIDAR+mono	**80.24**	**62.43**	**55.93**

New approaches regarding the base line improve mainly the easy mode (from 2% to 7%) and are minor for the moderate one (from 1.5% to 2%) and hard one (from 1% to 2%). ERF+YOLOv3 performs better than the Mask-RCNN for easy mode and hard mode but is a little worse for the moderate one.

E. Performance of BEV Vehicle Detection

Table III shows AP of bird's-eye view on the KITTI validation set using a IoU of 0.5. Results are quite similar to those obtained for 3D, because projection of the 3D bounding boxes in the ground plane is evaluated. In this case, a 2D IoU metric evaluates horizontal localization performance in the 3D driving space, a key parameter in autonomous driving. Objects' height is not taken into account, being the reason why numbers are a little higher than for the 3D case.

TABLE III: BEV detection performance: Average Precision (AP) in % of bird's eye view boxes on KITTI validation set for IoU=0.5

Method	Data	Easy	Mod.	Hard
Mono3D [8]	Mono	30.5	22.39	19.16
3DOP [19]	Stereo	55.04	41.25	34.55
VeloFCN [16]	LIDAR	79.68	63.82	62.80
MV3D [11]	LIDAR+mono	**96.52**	**89.56**	**88.94**
Our ERFNet [14]	LIDAR+mono	79.77	65.76	63.14
Our ERF+YOLOv3	LIDAR+mono	**89.85**	**75.16**	**66.49**
Our Mask-RCNN	LIDAR+mono	**86.93**	**75.62**	**68.63**

LIDAR-based methods perform better than those based on vision. The best results are obtained by MV3D because they use 2D/3D box regression from two different sensors instead of only a 2D box regression, as in vision methods, or a 2D/3D box regression from only one sensor, as in LIDAR methods. Regarding our fusion proposals, they perform better than LIDAR-based and vision-based methods but worse than MV3D. Improvements regarding the baseline are significant, being between 7% and 10% for the easy mode, about 10% for the moderate mode and between 3% and 5% for the hard mode. ERF+YOLOv3 performs better than Mask-RCC for the easy mode, has similar results for the moderate case and presents an opposite behaviour for the hard one.

Our final goal is to provide a good vehicle localization to our autonomous navigation system, and BEV metric is the

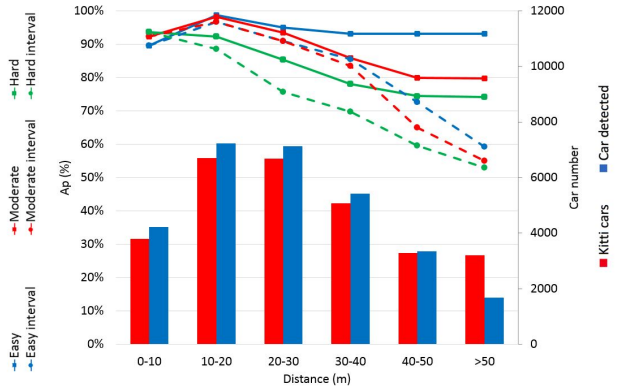


Fig. 2: AP of BEV boxes on KITTI validation set as a function of distance.

best fitted to this problem. KITTI evaluation is a way to show that our proposals are competitive regarding other state-of-the-art methods, but comparison should be taken with caution because our CNNs have not been trained with KITTI images and validation is carried out over the whole training set and not a subset of it. Besides, KITTI gives a mean AP for all vehicles, but AP changes with distance and not all detections have the same importance from a safety point of view. Thus, the detection of closer objects is more critical than that of more distant objects.

Fig. 2 shows an AP graph for the "car" class detected within a 10-meter range along distance. It also shows the number of ground truth/detected cars for each range. As can be seen, detection values are high for short distances and decrease with distance for the three modes. The detection range is above 90% for easy and moderate modes up to 30 meters, excluding cars that are very close and not fully detected in the image or in the LIDAR point cloud. These detection percentages, adding a tracking stage, can be suitable for an autonomous vehicle application such as ours in which the vehicle must travel at a maximum speed of 50 km/h in an urban environment.

F. Comparison of the New Framework Proposals

New proposals outperform results obtained with the baseline in the three tests carried out: 2D detection on images, 3D detection and BEV detection on the 3D space. For 2D detection, Mask-RCNN gets better numbers than ERF+YOLOv3. For 3D and BEV detection numbers are quite similar. If we take into account only the accuracy, our experiments suggest that using Mask-RCNN is the best one of the two approaches. However, if we consider the processing times of both approaches and bear in mind the computational constraints of a real vehicle, then we must choose the YOLOv3-based option as the most suitable option for our application. For 1024 x 720 images and using a 1070 NVIDIA GPU, YOLOv3 takes 50 ms and Mask-RCNN more than 2 s. In consequence, the ERF+YOLOv3 approach is the most appropriate for our real application.

G. Qualitative Results

Qualitative results on KITTI are provided in Fig.3. The BEV and image pairs and detected 3D bounding boxes by our ERF+YOLOv3 approach are shown. Our proposal detects the car quite well, even when the car is distant, heavily occluded or with different orientation. These results show the good scalability and generalization domain of our proposal. While LIDAR suffers from a high data sparsity for distant object detection, high resolution of images provides very useful information. Furthermore, we don't need a training stage with costly annotations because our proposal is based on pre-trained CNNs.

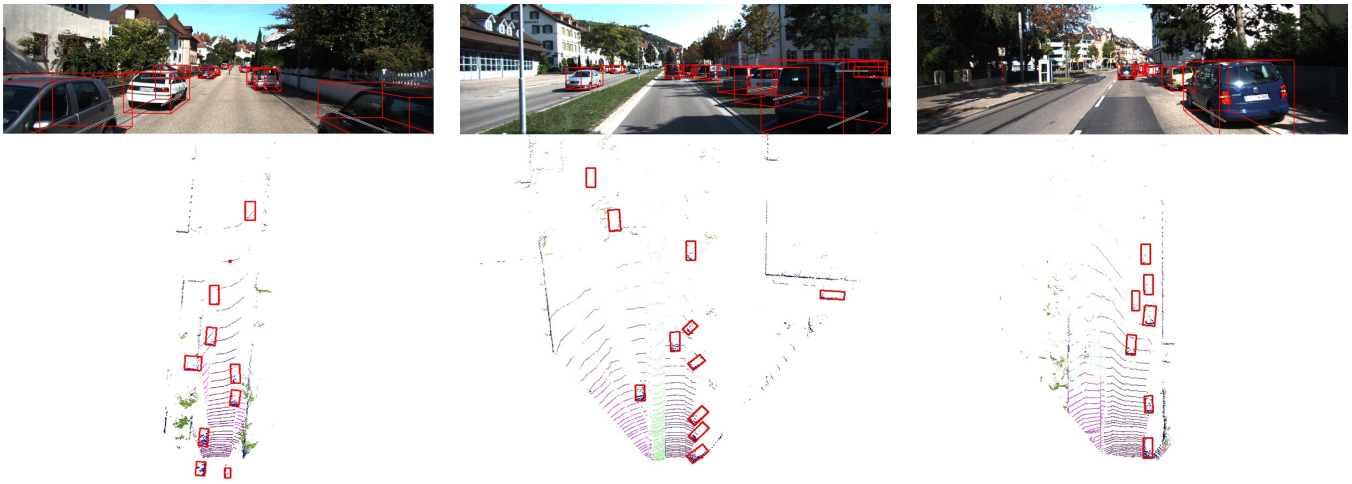
H. Real Autonomous Driving Application

We have evaluated our ERFNet+YOLOv3 framework in our autonomous electric car prototype, which is equipped with a Velodyne LIDAR (VLP-16) placed on top of the vehicle, which provides 16 channels of 360º horizontal FOV and +- 15º vertical FOV, and a ZED stereo camera, which provides 30 fps with a 1280x720 pixel resolution. For evaluation purposes we have obtained a small dataset with 200 frames extracted from 10 sequences recorded in the Campus of the University of Alcala. BEV bounding box annotations have been manually obtained over 70 meters for 1462 vehicles. Fig.4 shows an AP graph for vehicles detected within a 10-meter range, similar to those shown in Fig.2 for KITTI. As we can see, detection values are 100% until 20 m, over 96% until 30 m and decrease for longer distances. These results, adding a tracking stage, can be enough for our autonomous vehicle application (speed below 50 km/h) and show the generalization potential of our proposal because it has been run without any adaptation with respect to the configuration used for KITTI.

V. Conclusions and Future Works

In this paper we have presented two new approaches to detect vehicles in images and in a 3D and BEV scene, integrating two state-of-the-art CNNs such as YOLOv3 and Mask-RCNN in a previous framework, developed by the authors, that combine 3D LIDAR point clouds and semantic segmentation. Our methods take advantage of multimodal fusion, geometrical constraints, and pretrained modules to avoid an end-to-end learning. The new proposals have been tested using the KITTI object detection benchmark, showing some improvement with respect to the baseline and a good capability of generalization to any domain. Performance results are similar to other competitive state-of-the-art approaches in KITTI, but our architectures were pretrained on generalistic datasets, removing the need to train to a specific dataset, such as KITTI, and avoiding the costly efforts of annotating new training data. Furthermore, we have tested our proposal in a new driving environment as ours without any adaptation obtaining results suitable for the navigation of our autonomous car.

As future work we plan to test in depth the presented perception system in our autonomous navigation architecture, based on the Robot Operating System (ROS), over our electric prototype.



(a) *Image 000025* **(b)** *Image 000038* **(c)** *Image 000046*

Fig. 3: Quantitative results on KITTI dataset: 2D boxes in images, 3D boxes in 3D space and 3D projected boxes in the BEV

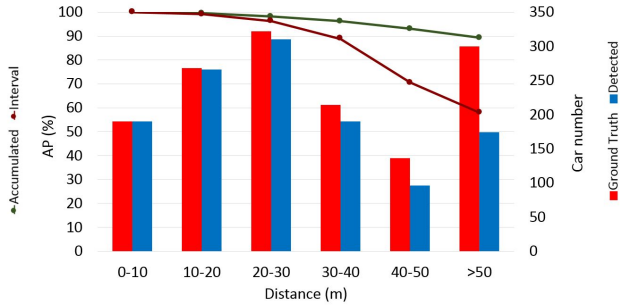


Fig. 4: AP of BEV boxes on Campus validation set as a function of distance.

REFERENCES

[1] R. Barea, C. Pérez, L. M. Bergasa, E. López-Guillén, E. Romera, E. Molinos, M. Ocana, and J. López, "Vehicle detection and localization using 3d lidar point cloud and image semantic segmentation," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3481–3486, IEEE, 2018.

[2] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 641–656, 2018.

[3] S.-L. Yu, T. Westfechtel, R. Hamada, K. Ohno, and S. Tadokoro, "Vehicle detection and localization on bird's eye view elevation images using convolutional neural network," in *Safety, Security and Rescue Robotics (SSRR), 2017 IEEE 17th International Symposium on*, pp. 102–109, IEEE, 2017.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361, IEEE, 2012.

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[8] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2156, 2016.

[9] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 1513–1518, IEEE, 2017.

[10] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018.

[11] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *IEEE CVPR*, vol. 1, p. 3, 2017.

[12] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *arXiv preprint arXiv:1712.02294*, 2017.

[13] E. Romera, L. M. Bergasa, J. M. Alvarez, and M. Trivedi, "Train here, deploy there: Robust segmentation in unseen domains," in *Proceedings of the IEEE conference on Intelligent Vehicles Symposium, p. to appear, IEEE ITS*, 2018.

[14] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.

[15] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection.," in *Robotics: Science and Systems*, vol. 1, p. 5, 2015.

[16] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *arXiv preprint arXiv:1608.07916*, 2016.

[17] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[19] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, pp. 424–432, 2015.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[21] B. J. M. D. Guindel, C. and F. García, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[22] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[23] P. D. K. He, G. Gkioxari and R. Girshick, "Maskr-cnn," in *Proceedings of the IEEE International Conference on Compueter Vision (ICCV)*, 2017.